



Fermilab

AP-Note-90-011

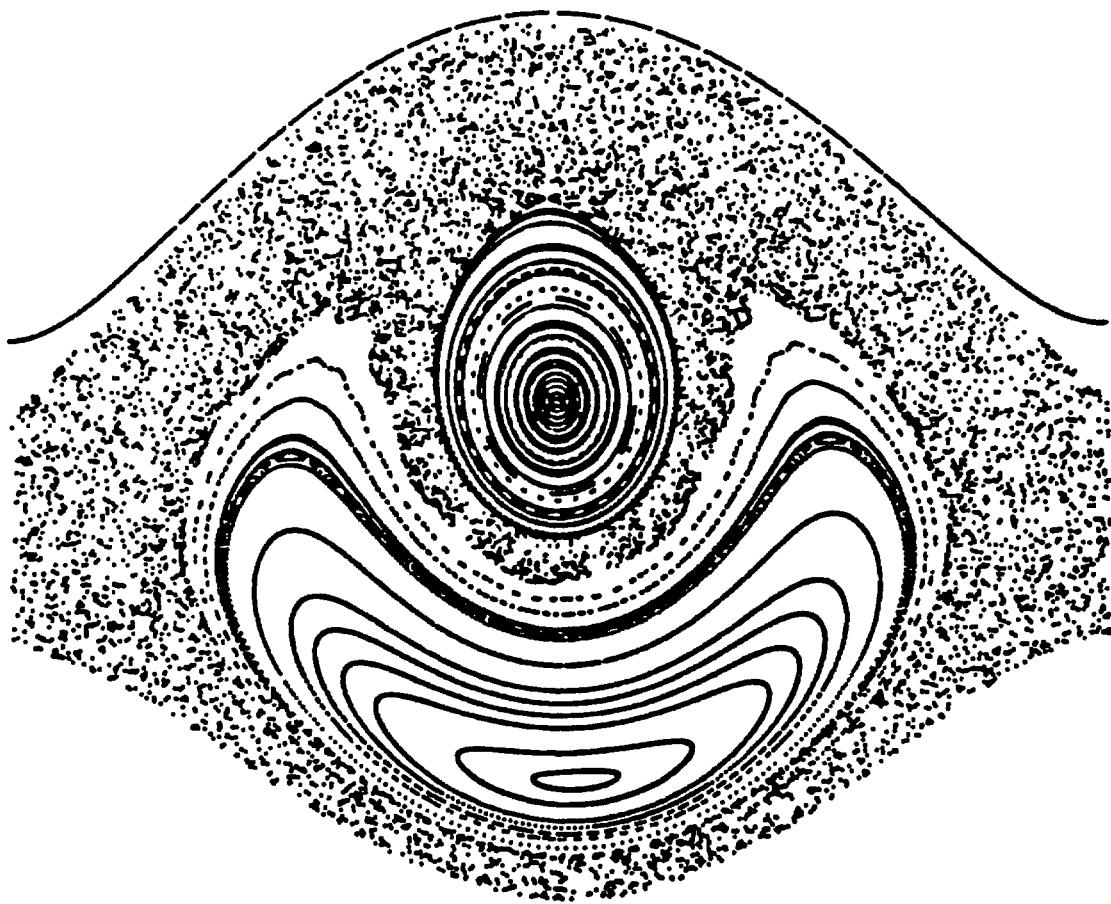
A
C
C
E
L
E
R
A
T
O
R

P
H
Y
S
I
C
S
D
E
P
A
R
T
M
E
N
T

TITLE: Unix Data Acquisition System:
A Short User Manual

AUTHOR: Ioannis Kourbanis and George P. Tsironis

30 August 1990



Unix Data Acquisition System: A Short User Manual

Ioannis Kourbanis and George P. Tsironis

Fermi National Accelerator Laboratory
Batavia, IL 60510

30 August 1990

The Unix Data Acquisition System (UDAS) is a Fermilab based prototype for on line data acquisition and analysis. It relies on the harmonious interfacing of real data with computer codes written in the C language. The system employs a shared memory approach that has been previously introduced and successfully tested in the context of the E778 nonlinear beam dynamics experiment. In its current initial experimental implementation UDAS relies on the codes *ruby* for data analysis and *kaspar* for graphics. The present note, which is primarily addressed towards potential users, describes briefly the current UDAS implementation in the Fermilab Main Ring (MR) and gives step-by-step directions for its use. A more complete manual will be written after the anticipated hardware and especially software upgrade that will be taking place during the following months.

At present the UDAS system works as follows: Two beam position monitors (bpm's) located near the F4 service building collect MR turn-by-turn data through the cycle. Bpm F43 senses the vertical and F44 the horizontal position of the beam. These data are digitized by a LeCroy 8610 waveform digitizer that in turn passes the data to *fig*. The latter is a 3/60 SUN workstation that accumulates the data in a shared memory location. Real time data analysis is possible after the data are passed from *fig* to *myrtle*, another 3/60 SUN workstation that is located in the control room annex. Most of the data handling can be done simply by using the middle button of the workstation's mouse. Various X window commands are available using the

settings, he must change the defaults in file *bpma2*. Shortly, this will be handled automatically through the use of a *camel*, i.e. a CAMac panEL. In Fig. 1 you see how the screen looks right after the above steps have been completed and *gocam* is running.

After these preliminary steps you can now look at turn-by-turn data, do fft's for selected data segments and obtain vertical and horizontal tunes. The segment selection is done in the **Tune_Window** and all the rest through the use of the middle mouse button. To see the turn-by-turn data use the **middle** button and press

- **ship a to myrtle** in order to bring data from **fig** to **myrtle**
- **start kaspar a** to see the data

The data from the vertical bpm are displayed above and the ones from the horizontal below, both as a function of turn number. These are data through the selected MR cycle (Fig. 3). You can use the *kaspar* panel to zoom into a segment of the data (Figs. 4 and 5). You can zoom into the data by selecting a starting point on the plot window and drag the mouse while depressing the left or middle button. You can go back to the original coordinate limits by clicking the diamond on the upper right hand corner of the window. You should experiment with *kaspar* and discover its powerful data handling capabilities.

Let us now fft the turn-by-turn data at injection (Fig. 4 or 5). In our example the relevant segment starts approximately at turn number 3165 and lasts for about 100 turns. First go to **Tune_Window** and type

- **b, v, or h**
- **3165**
- **100**

This is input for the program *tuneout* that, in turn, feeds these numbers into *ruby* which starts running next, when you click the middle button and select

- **V-and-H TUNES** for both vertical and horizontal fft of this segment,
- **display Vtune** for only vertical or

In Fig. 10 we show how the myrtle console looks after a typical sequence. Note that you can move the windows around by pressng the right button while the pointer is located anywhere on the window you want to move. Also, you can change the dimensions of a window by placing the pointer on the small square on the upper right hand corner of that window and dragging it along.

References

- [1] Rod Gerig, Quentin King, Steve Peggs and George Tsironis, *UDAS Proposal - a Unix Data Acquisition System*, Fermilab, February 1990.

| | M4 PINGER, BPM TIMERS | D/A | REF | Eng-U ♦COPIES♦ |
|------------------|--------------------------|--------------------|-------------------|----------------|
| -<FTP>+ *LX♦ | X=A/D X=TIME | Y=M ERING ,M:RFSUM | ,M:PSHIFT,M:M3APG | |
| COMMAND BL-- | Eng-U I= 0 | I= 0 | ' 0 | ' -200 , 0 |
| -< 4>+ S_MR AUTO | F= 5 | F= 150 | , 5 | , 200 , 30 |
| -M:MRSTV1 | MNT RNG SWP TRIG VERNIER | 2 | VOLT | |
| -M:E48PN | E48 Pinger Setting | .096 | KV | *T.H |
| M: E48PHV | E48 Pinger Reading | | | |
| -T:PINGT | PINGT TUNE PINGER TI | .2 | <29/ | / / > * |
| -T:E7781 | E778 arm trigger #1 | .05 | <29/ | / / > :: |
| -T:E7782 | E778 arm trigger #2 | .1 | <29/ | / / > :: |
| -M:BDOF29 | SLOW Damp OFF @ \$29 | 1.671 | <29/ | / / > :: |
| -M:MRVART | MAIN RING VARIABLE TR | .35 | <21/29/ | / / > :: |
| -G:SMRSTD | Misc:Start MR Studies | .13 | <29/ | / / > :: |
| -M:BINJOF | DB-Bend injection off | -51.83151 | 4MeV | |
| -M:A1BP1 | PREP FOR TURN STUDIES | .158 | <20/21/29/2A/2B♦ | :: |
| -M:A1BP2 | ENABLE TURN STUDIES | .198 | <20/21/29/2A/2B♦ | :: |
| -M:A2BP1 | PREP FOR TURN STUDIES | .06 | <20/21/29/2A/2B♦ | *. |
| -M:A2BP2 | ENABLE TURN STUDIES | .1 | <20/21/29/2A/2B♦ | * |
| -M:BPMPFB | MR: BPM Prepare for Be | .001 | <29/ | / / > ::- |
| -M:BPMBUN | MR: BPM Set Bunch Mode | .15 | <20/2A/2B/ | / > :: |
| -M:BPDF20 | BPM Display Frame - | \$ 7.85 | <20/ | / / > * |
| -M:BPDF21 | BPM Display Frame - | \$ 3.8 | <21/ | / / > :: |
| -M:BPDF29 | BPM Display Frame - | \$ 2 | <29/ | / / > * |
| -M:BPDF2A | BPM Display Frame - | \$.4 | <2A/ | / / > * |
| -M:BPDF2B | BPM Display Frame - | \$.295 | <2B/ | / / > * |
| -M:BPDF2D | BPM Display Frame - | \$.15 | <2D/ | / / > * |
| -M:BPDF2E | BPM Display Frame - | \$.3 | <2E/ | / / > * |

Figure 2

Figure 4

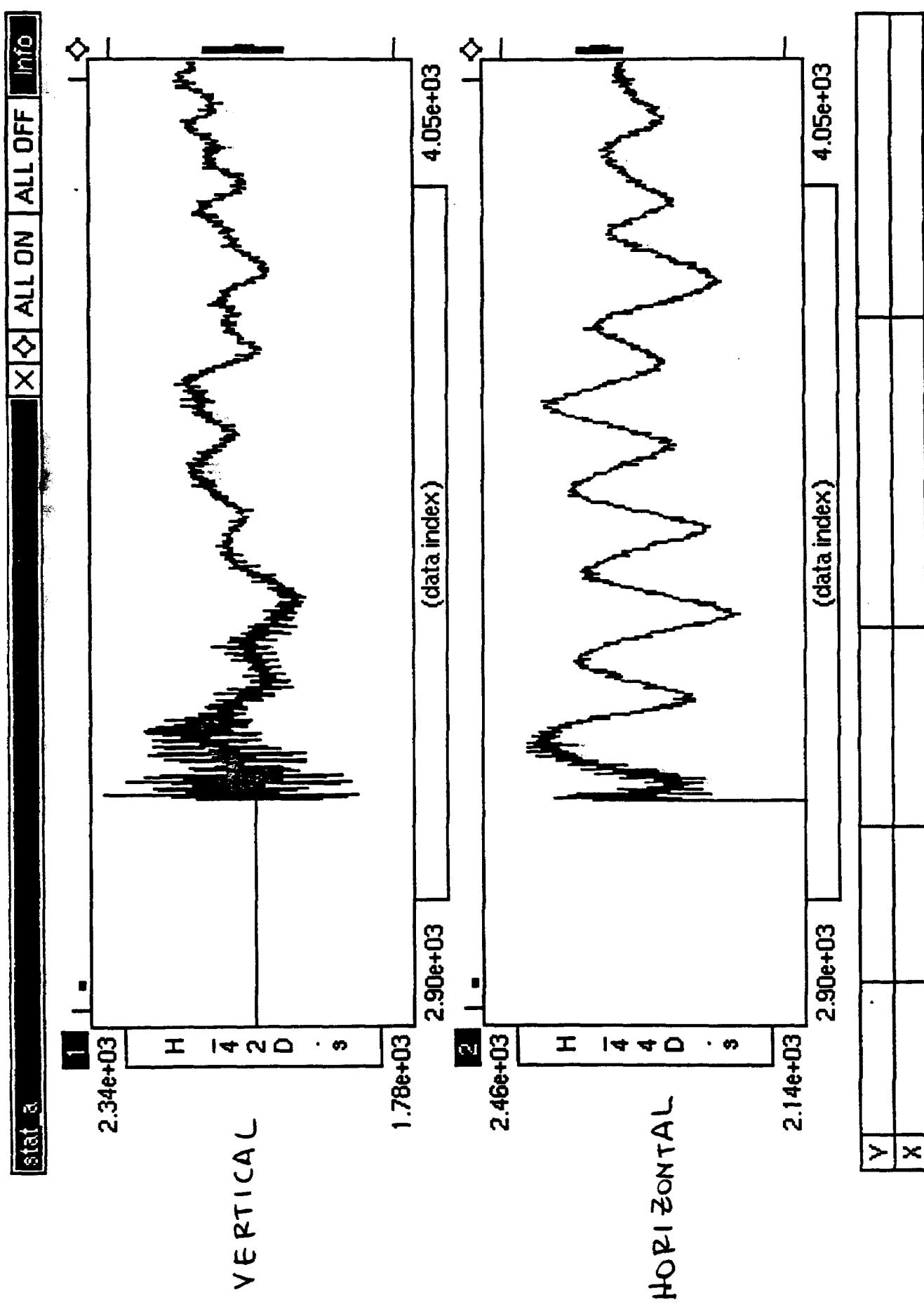


Figure 6

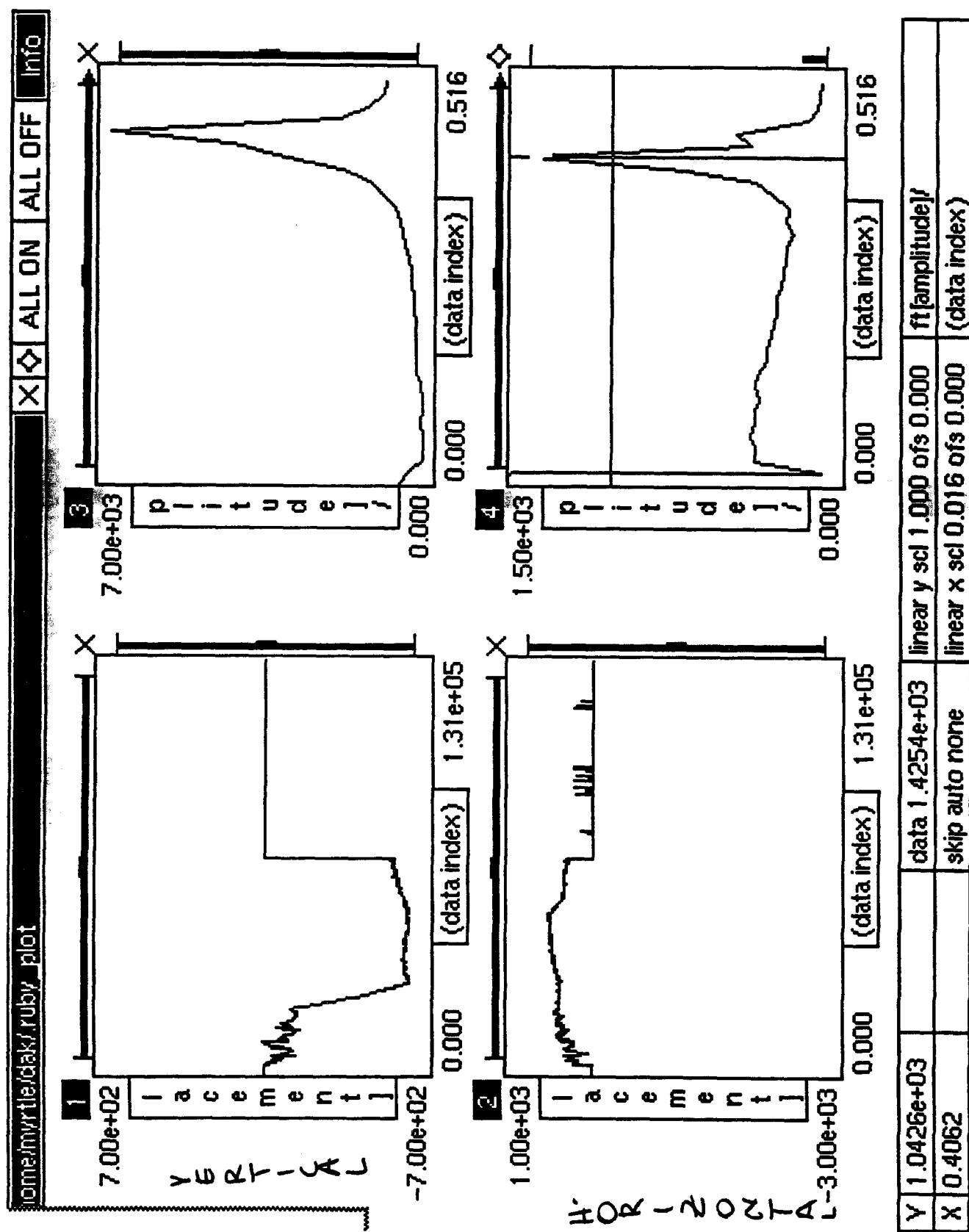


Figure 8

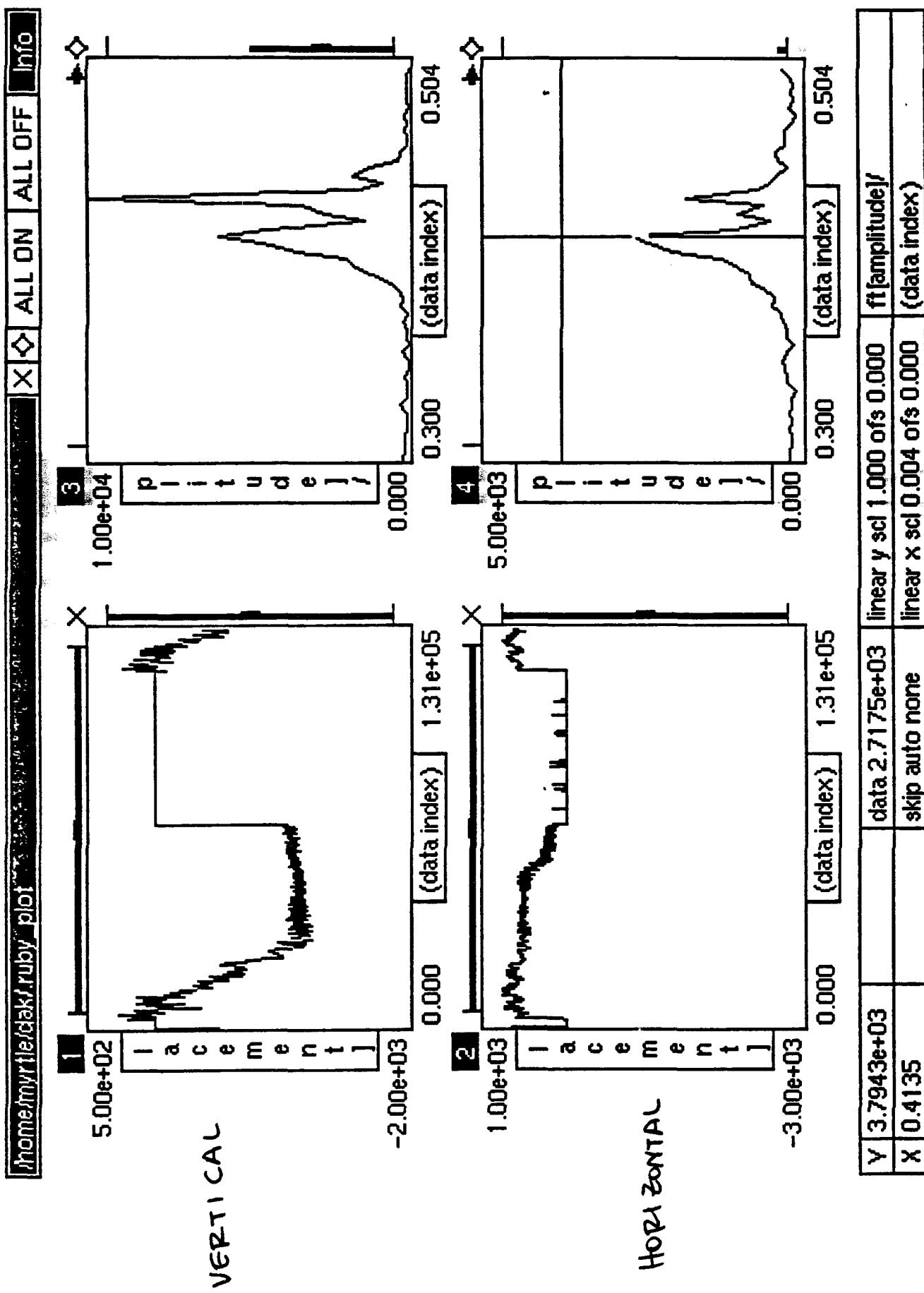


Figure 10

